# zmock介绍

## zmock功能

模拟出一个不存在的http接口，根据请求参数的不同返回对应的数据。

## zmock应用场景

1. 在接口文档写好后，就可以mock出一个接口，以便于客户端开发，且测试人员也可以开始写测试用例。mock出来的接口也相当于一种契约，服务端与客户端都必须遵守这个mock出来的契约。
2. 当系统中有第三方接口，在做性能测试时，可以mock第三方接口，作为性能测试的档板。
3. 其它。

## zmock介绍

1. 开发语言及框架：java，spring，easyui
2. web容器：tomcat
3. 部署方式：

* 将打包好的zmock.war更名为ROOT.war
* 将ROOT.war放在tomcat的webapps目录下面
* 启动tomcat

1. 设置

* 打开<http://ip:port/zmock/index?role=whosyourdaddy>
* 填写mock数据保存目录，作为一个workspace。在<https://git.oschina.net/pinghailinfeng/nbmock中有一个data目录>，建议将mock数据保存目录指向这个data目录，里面有已经写好的mock接口，可以作为参考。
* 一般情况下用http://ip:port/zmock/index访问即可

## zmock平台使用说明

* 接口文档：

请求链接：/api/info/v1

请求方法：post

请求参数示例：{"name":"zhang","age":30}

请求参数说明：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数类型 | 是否必需 | 示例值 |
| name | string | 是 | zhang |
| age | int | 是 | 30 |

响应说明：

1. 当name的值为空或者age的值小于等于0时，响应：  
   {"retCode":"300","retMsg":"参数不正确"}
2. 当name的值为zhan时，响应:  
   {"retCode":"301","retMsg":"数据不存在"}
3. 当name值为zhang且age的值为18时，响应：

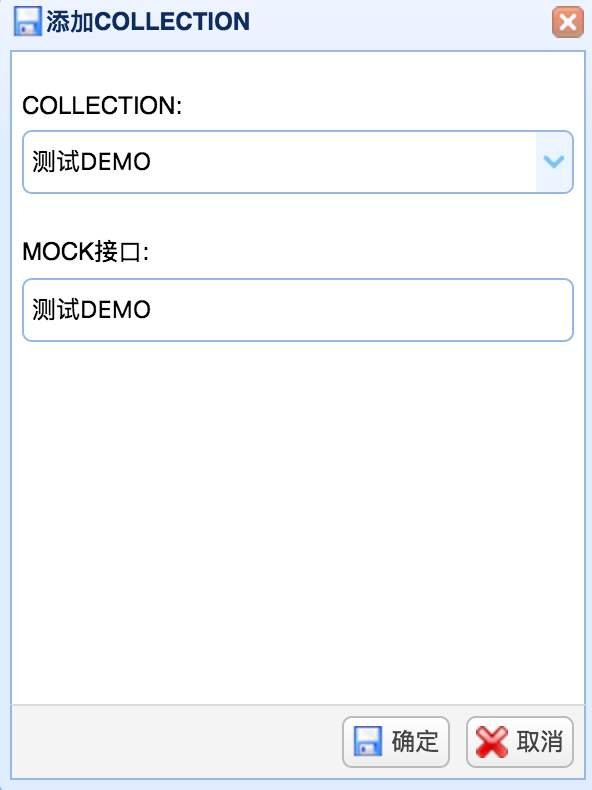
{"retCode":"301","retMsg":"数据不存在"}

1. 其它情况下，响应：

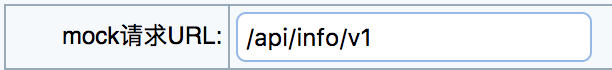
{"retCode":"200","retMsg":"ok.","data":{}}

* 以上是一份简单的接口文档，公司不同，文档的要求不同，但我相信大家都应该能看懂这份文档，现在，我们根据这份文档来mock一个http接口出来：

1. 点击左上角的+号：



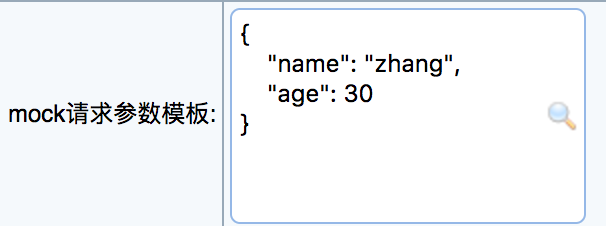
1. 确定后，在菜单中选择或输入测试DEMO，在mock请求URL中填写：



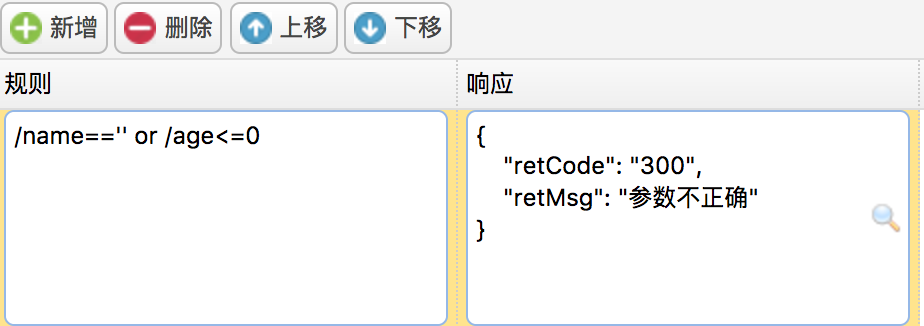
1. mock请求方法选择POST:



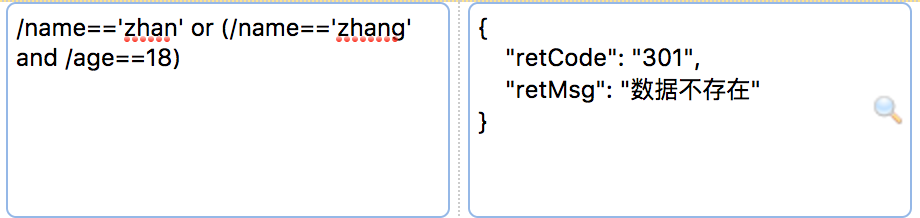
1. 在mock请求参数模板中：



1. 在mock规则定义中点击新增：



1. 再增加一条规则：

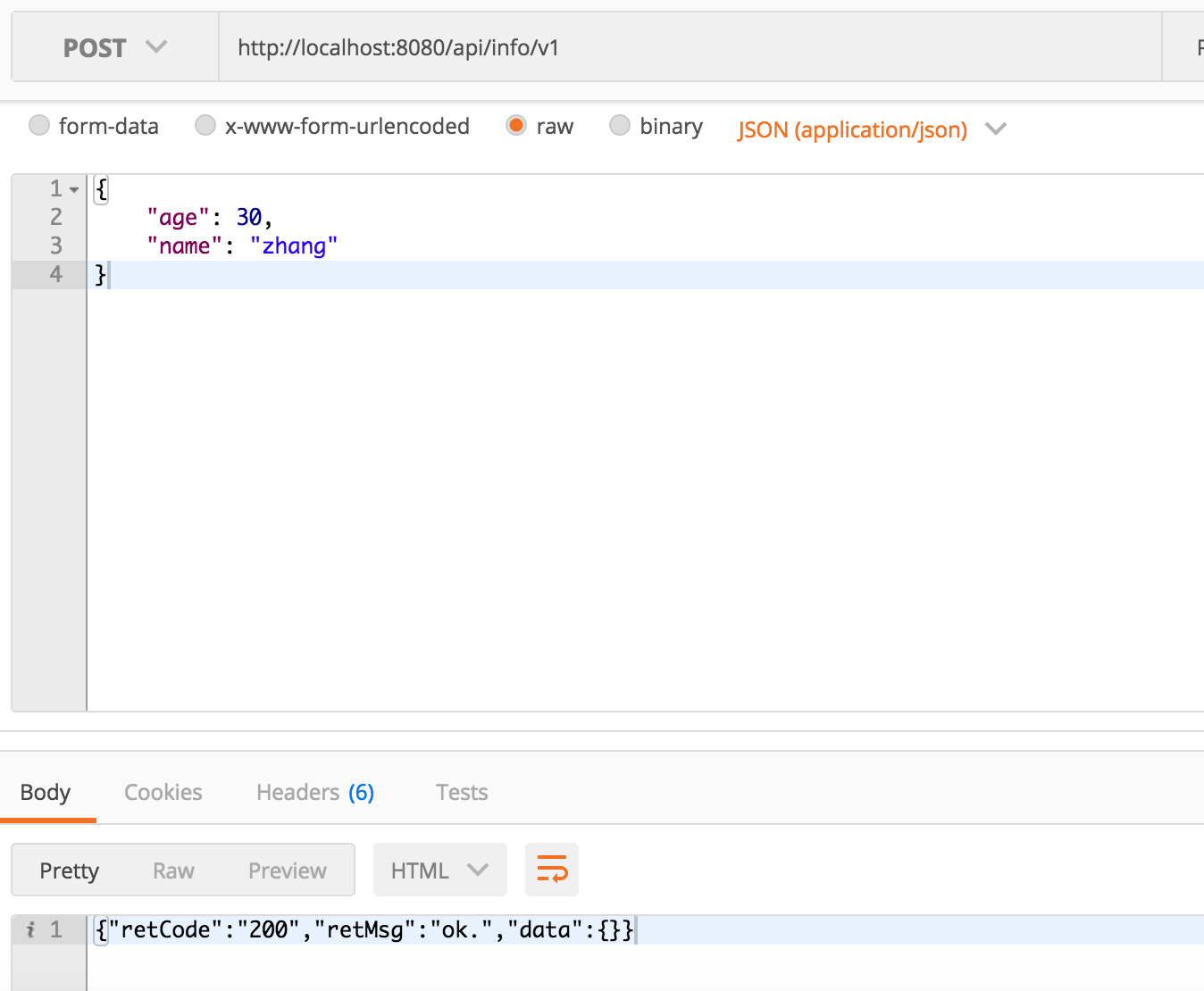


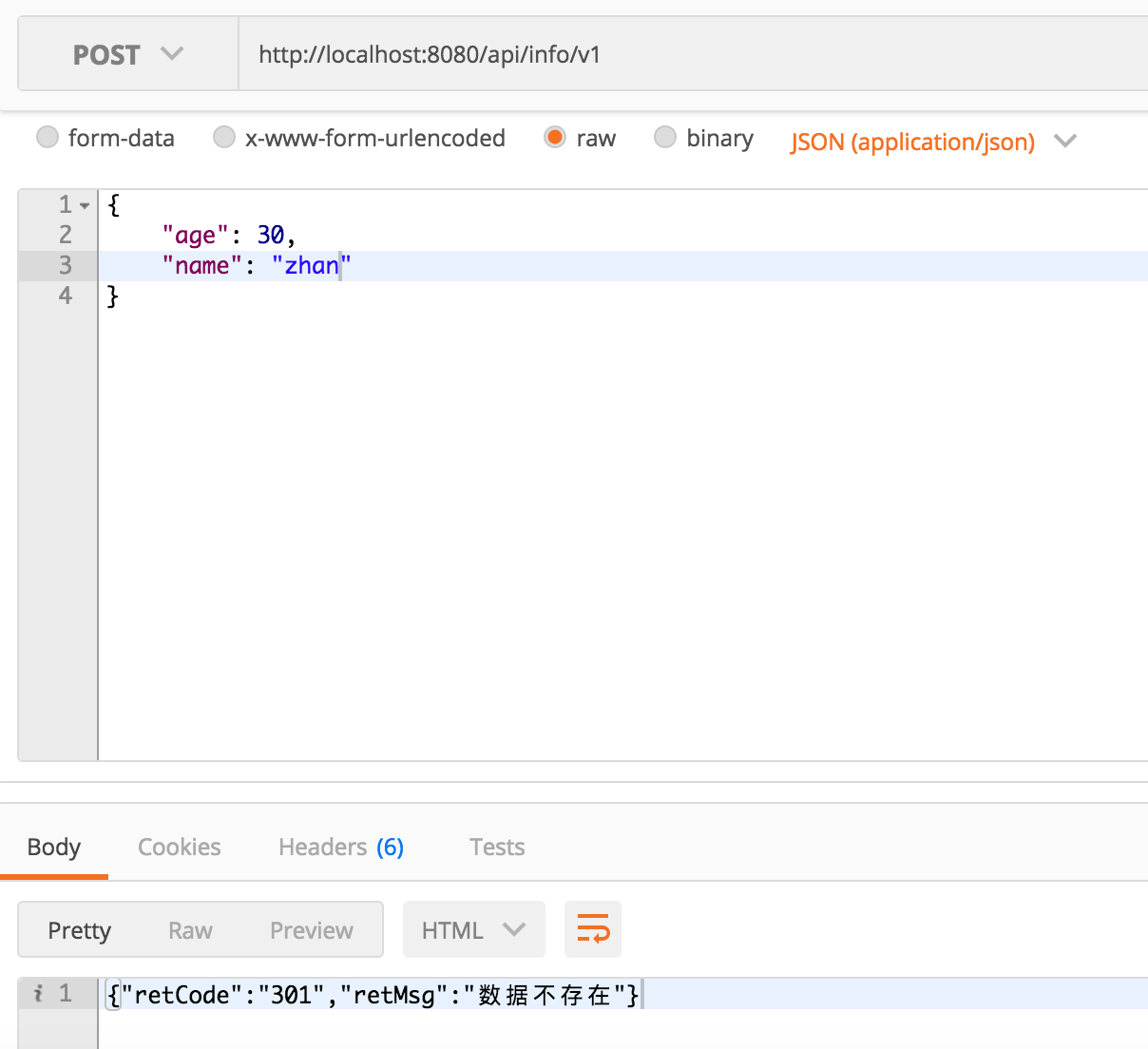
1. 最后一条规则：

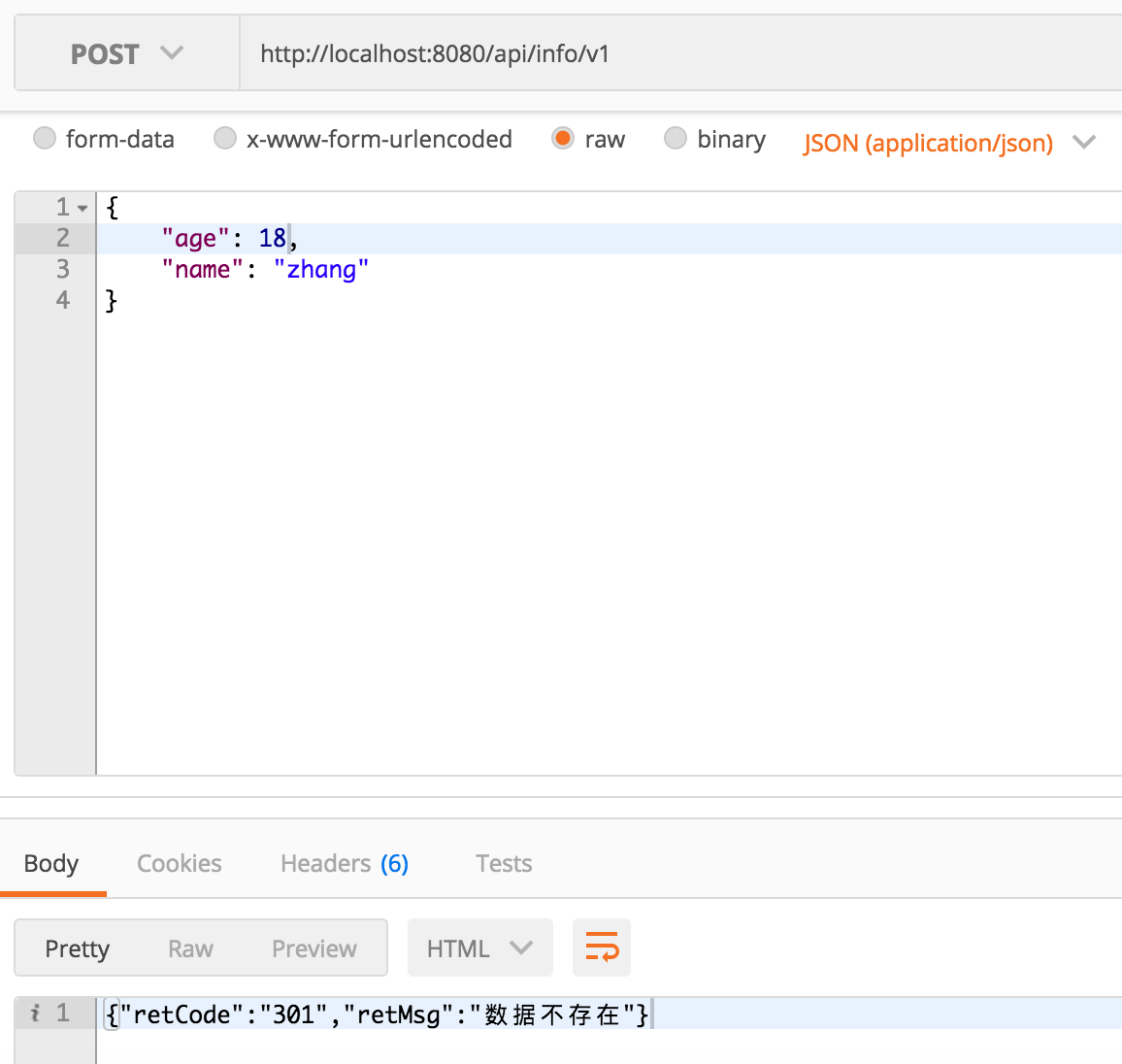


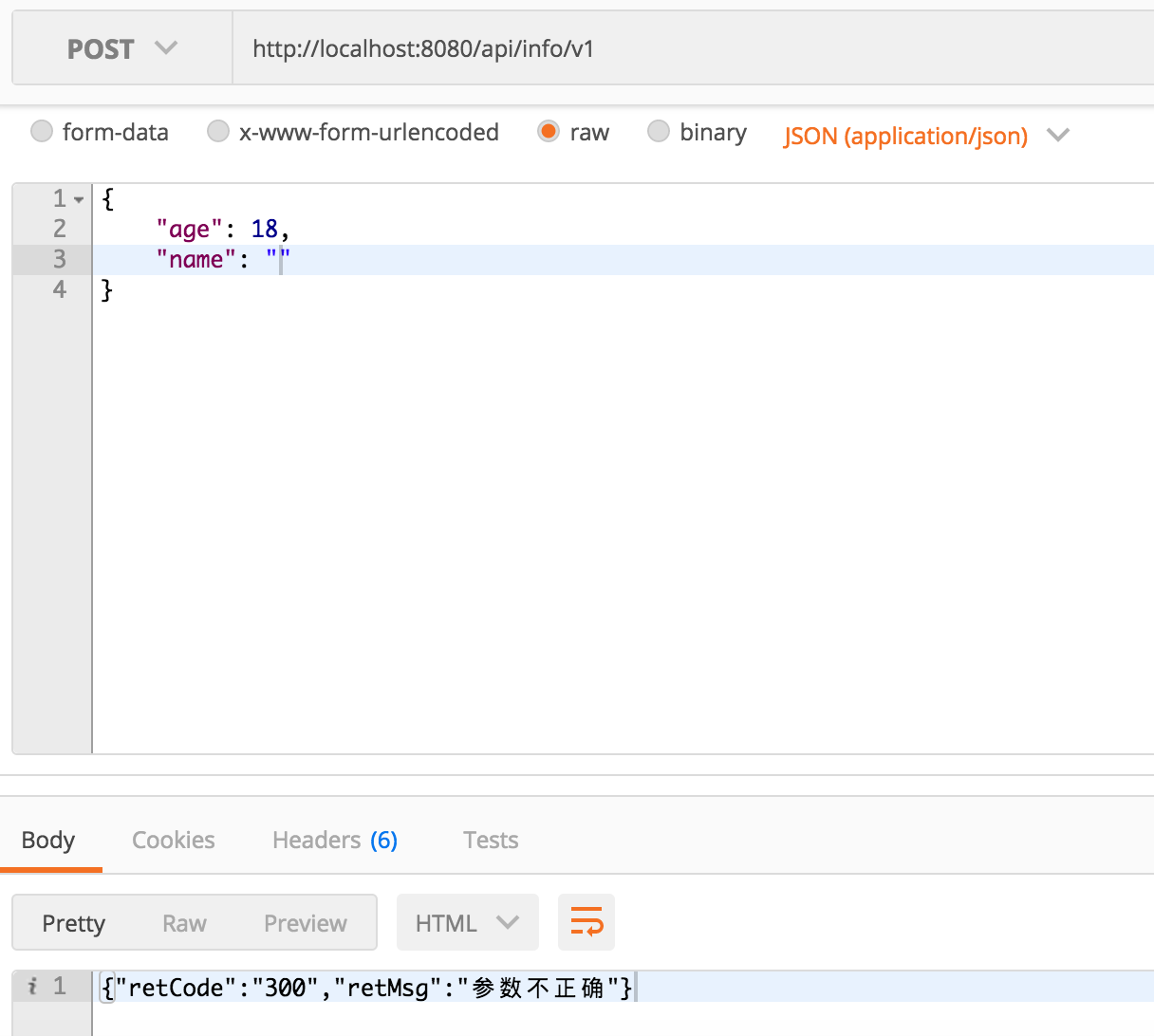
1. 保存后，mock就完成了。

* mock完成后，我们来测试一下这个http接口：









## zmock的特点

1. 请求参数模板的定义。根据请求参数模板，来判断请求参数是否正确。

mock请求参数模板说明:

* .\*表示请求参数为任意字符串，用于POST BODY
* 如username=&password=表示POST表单提交或GET请求
* 如{"username":"","password":""}表示POST BODY为JSON串

1. mock规则利用表达示来定义。比如/name,/age，这是json路径，具体的请参考：https://github.com/zhangfei19841004/zson

表达示所支持的比较符有：

== != > >= < <= in contains

表达示方便扩展，如有需要，自行扩展即可。

支持的连接符有：and or (),其中()的优先级最高，支持括号里套括号。

1. 采用文件异步保存数据的方式，且数据都放在内存里，保证接口的响应速度。